UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/772,698	02/05/2004	Gansha Wu	ITL.1069US (P18137)	4898

21906 7590 12/13/2006

TROP PRUNER & HU, PC
1616 S. VOSS ROAD, SUITE 750
HOUSTON, TX 77057-2631

EXAMINER	
SYED, FARHAN M	
ART UNIT	**PAPER NUMBER**
2165	

DATE MAILED: 12/13/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/772,698	Applicant(s) WU ET AL.	
	Examiner Farhan M. Syed	Art Unit 2165	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 28 September 2006.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-28 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-28 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 05 February 2004 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-28 are pending.

Response to Remarks

Claim Rejections - 35 USC § 112

2. Applicant's arguments, see pages 6 and 7, filed 28 September 2006, with respect to claims 13, 18-24, and 26-28 have been fully considered and are persuasive. The rejection of claims 13, 14-24, and 26-28 has been withdrawn.

Claim Rejections - 35 USC § 101

3. Applicant's arguments, see page 7, filed 28 September 2006, with respect to claims 1, 18-24, and 26-28 have been fully considered and are persuasive. The rejection of claims 1, 14-24, and 26-28 has been withdrawn.

Response to Argument

4. Applicant's arguments with respect to claims 1-28 have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1, 2, 5, 7, 10, 12, 13, 14, 16-18, 21, 23, 24, 27, and 28 are rejected under 35 U.S.C. 102(e) as being anticipated by Boger et al (U.S. Patent 6,996,556 and known hereinafter as Boger) in view of a non-patent literature titled "Web Mining for Web Personalization" by M. Eirinaki et al., ACM Transactions on Internet Technology, Vol. 3, No. 1, February 2003, pages 1-27 (and known hereinafter as Eirinaki).

As per claims 1, 18, and 24, Boger teaches a method comprising: storing method metadata (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata to a plurality of optimizer instances in connection with the optimization of database queries." "As an additional matter, by implementing metadata handling functionality outside of optimizer instances, improvements to metadata collection, refinement and maintenance algorithms may be implemented within a metadata manager without requiring modification of optimizer program code."* The preceding text clearly indicates that method metadata are a collection of metadata, a cookie indicator is the optimizer, and code portion is an instance of a program code.)(Column 4, lines 50-54; lines 64-67; column 5, lines 1-2)

However, Boger does not teach the method including a cookie indicator in a code portion; and storing the code portion, in a memory, for later use.

Eirinaki teaches a method including a cookie indicator in a code portion (i.e. *"Web site personalization can be defined as the process of customizing the content and structure of a Web site to the specific and individual needs of each user taking advantage of the user's navigational behavior. The steps of a Web personalization process include: (a) the collection of Web data, (b) the modeling and categorization of these data (preprocessing phase), (c) the analysis of the collected data, and (d) the determination of the actions that should be performed. The ways that are employed in order to analyze* *the collected data include content-based filtering, collaborative filtering, rule-based filtering, and Web usage mining. The site is personalized through the highlighting of existing hyperlinks, the dynamic insertion of new hyperlinks that seem to be of interest for the current user, or even the creation of new index pages."* According to Mircosoft's Computer Dictionary, 5th ed., a cookie is a block of data that a Web server stores on a client system. When a user returns to the same Web site, the browser sends a copy of the cookie back to the server. Cookies are used to identify users, to instruct the server to send a customized version of the requested Web page, to submit account information for the user, and for other administrative purposes. Similarly, the Web personalization exemplifies this concept by customizing the content and structure of a Web site to the specific and individual needs of each user taking advantage of the user's navigational behavior. It is clearly obvious that to accomplish web personalization, a cookie indicator must exist in the code portion. See also section 3 of user profiling.)(Pages 3-9, section 2); and storing the code portion, in a memory, for later use (i.e. *"In order to personalize a Web site, the system should be able to distinguish between different users or groups of users. This process is called user profiling and its objective is the creation of an information base that contains the preferences, characteristics, and activities of the users. In the Web domain and especially in e-commerce, user profiling has been developed significantly because Internet technologies provide easier means of collecting information about the users of aWeb site, which in the case of e-business sites are potential customers. A user profile can be either static, when the information it contains is never or rarely altered (e.g., demographic information), or dynamic when the user profile's data change frequently. Such information is obtained either explicitly, using online registration forms and questionnaires resulting in static user profiles, or implicitly, by recording the navigational behavior and/or the preferences of each user, resulting in dynamic user profiles. In the latter case, there are two further options: either regarding each user as a member of a group and creating aggregate user profiles, or addressing any changes to each user individually. When addressing the users as a group, the method used is the creation of aggregate user profiles based on rules and patterns extracted by applying Web usage mining techniques to Web server logs. Using this knowledge, the Web site can be appropriately customized. This case is* *discussed in detail in Section 4, therefore it won't be further analyzed here. In the following sections, we provide a description of several methods for implicit and explicit collection of user profile data. Privacy issues that arise in the user profiling process are discussed, and an overview of available tools and user profiling applications is presented."* Based on the previous analysis above, it is clearly obvious that a storing a code portion in the cookie indicator is used to determine the web personalization of user web content.)(Pages 5-9, section 3).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Erinaki to include the method including a cookie indicator in a code portion; and storing the code portion, in a memory, for later use with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claim 2, Boger teaches a method, wherein the method metadata further comprises a method handle (i.e. *"As discussed hereinafter, a metadata manager may be configured to handle the tasks of collecting, refining, retrieving and/or maintaining metadata for one or more databases, thus off-loading responsibility for such tasks from individual optimizers or optimizer instances that make use of that metadata."* It is clearly obvious that to handle the tasks of collecting, refining, retrieving and/or maintaining metadata for one or more databases, a method handle is used.)(Column 4, lines 54-59).

As per claim 5, Boger teaches a method, further comprising storing the method metadata at an end of the code portion (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata to a plurality of optimizer instances in connection with the optimization of database queries." "As an* *additional matter, by implementing metadata handling functionality outside of optimizer instances, improvements to metadata collection, refinement and maintenance algorithms may be implemented within a metadata manager without requiring modification of optimizer program code."*)(Column 4, lines 50-54; lines 64-67; column 5, lines 1-2).

As per claims 7 and 21, Boger teaches a method, further comprising querying the code portion for the method metadata (i.e. *"The primary task of a query optimizer is to choose the most efficient way to execute each database query, or request, passed to the database management system by a user." "An optimizer is often permitted to rewrite a query (or portion of it) into any equivalent form, and since for any given query there are typically many equivalent forms, an optimizer has a countably infinite universe of extremely diverse possible solutions (plans) to consider."*)(Column 1, lines 63-66; Column 2, lines 17-22).

As per claims 10, 23, and 27, Boger teaches a method, further comprising storing the method metadata between a first basic block and a second basic block of the code portion (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata to a plurality of optimizer instances in connection with the optimization of database queries."*)(Column 4, lines 50-54; lines 64-67).

As per claim 12, Boger teaches a method further comprising storing the method metadata in a basic block used for exception handling (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata to a plurality of optimizer instances in connection with the optimization of database queries."*)(Column 4, lines 50-54; lines 64-67).

As per claim 13, Boger does not explicitly teach a method comprising of receiving a request from a requestor to query a code portion for a method bundle including method metadata; searching the code portion for the method bundle; and returning the method bundle to the requestor.

Eirinaki teaches a method comprising: receiving a request to query a code portion for a method bundle including method metadata *(i.e. "The purpose of Web usage mining is to reveal the knowledge hidden in the log files of a Web server. By applying statistical and data mining methods to the Web log data, interesting patterns concerning the users' navigational behavior can be identified, such as user and page clusters, as well as possible correlations between Web pages and user groups. TheWeb usage mining process can be regarded as a three-phase process, consisting of the data preparation, pattern discovery, and pattern analysis phases [Srivastava et al. 2000]. In the first phase, Web log data are preprocessed in order to identify users, sessions, pageviews and so on. In the second phase, statistical methods, as well as data mining methods (such as association rules, sequential pattern discovery, clustering, and classification) are applied in order to detect interesting patterns. These patterns are stored so that they can be further analyzed in the third phase of the Web usage mining process. A description of the fields included in a log entry of a Web usage log follows, along with a set of definitions of Web data abstractions, such as Web site, user, session, pageviews, and clickstreams. Technical issues concerning data preparation are discussed. A more detailed analysis of the methods employed in the Web usage mining process including simple log analysis is presented. Finally, a brief overview of the commercially available tools and applications specializing in log analysis or Web usage mining is given. Log analysis is regarded as the simplest method used in the Web usage mining process. The purpose of Web usage mining is to apply statistical and data mining techniques to the preprocessedWeb log data, in order to discover useful patterns. As mentioned before, the most common and simple method that can be applied to such data is statistical analysis. More advanced data mining methods and algorithms tailored appropriately for use in the Web domain include association rules, sequential pattern discovery, clustering, and classification. Association rule mining is a technique for* *finding frequent patterns, associations, and correlations among sets of items. Association rules are used in order to reveal correlations between pages accessed together during a server session. Such rules indicate the possible relationship between pages that are often viewed together even if they are not directly connected, and can reveal associations between groups of users with specific interests. Aside from being exploited for business applications, such observations also can be used as a guide forWeb site restructuring, for example, by adding links that interconnect pages often viewed together, or as a way to improve the system's performance through prefetching Web data. Sequential pattern discovery is an extension of association rules mining in that it reveals patterns of cooccurrence incorporating the notion of time sequence. In theWeb domain such a pattern might be aWeb page or a set of pages accessed immediately after another set of pages. Using this approach, useful users' trends can be discovered, and predictions concerning visit patterns can be made. Clustering is used to group together items that have similar characteristics. In the context of Web mining, we can distinguish two cases, user clusters and page clusters. Page clustering identifies groups of pages that seem to be conceptually related according to the users' perception. User clustering results in groups of users that seem to behave similarly when navigating through a Web site. Such knowledge is used in e-commerce in order to perform market segmentation but is also helpful when the objective is to personalize a Web site. Classification is a process that maps a data item into one of several predetermined classes. In the Web domain classes usually represent different user profiles and classification is performed using selected features that describe each user's category. The most common classification algorithms are decision trees, na¨ıve Bayesian classifier, neural networks, and so on. After discovering patterns from usage data, a further analysis has to be conducted. The exact methodology that should be followed depends on the technique previously used. The most common ways of analyzing such patterns are either by using a query mechanism on a database where the results are stored, or by loading the results into a data cube and then performing OLAP operations. Additionally, visualization techniques are used for an easier interpretation of the results. Using these results in association with content and structure information concerning the Web site there can be extracted useful knowledge for modifying the site according to the correlation between user and content groups."*)(Pages 9-16, section 4); searching the code portion for the method bundle *(i.e. "The purpose of Web usage mining is to reveal the knowledge hidden in the log files of a Web server. By applying statistical and data mining methods to the Web log data, interesting patterns concerning the users' navigational behavior can be identified, such as user and page clusters, as well as possible correlations between Web pages and user groups. TheWeb usage mining process can be regarded as a three-phase process, consisting of the data preparation, pattern discovery, and pattern analysis phases [Srivastava et al. 2000]. In the first phase, Web log data are preprocessed in order to identify users, sessions, pageviews and so on. In the second phase, statistical methods, as well as data mining methods (such as association rules, sequential pattern discovery, clustering, and classification) are applied in order to detect interesting patterns. These patterns are stored so that they can be further analyzed in the third phase of the Web usage mining process. A description of the fields included in a log entry of a Web usage log follows, along with a set of definitions of Web data abstractions, such as Web site, user, session, pageviews, and clickstreams. Technical issues concerning data preparation are discussed. A more detailed analysis of the methods employed in the Web usage mining process including simple log analysis is presented. Finally, a brief overview of the commercially available tools and applications specializing in log analysis or Web usage mining is given. Log analysis is regarded as the simplest method used in the Web usage mining process. The purpose of Web usage mining is to apply statistical and data mining techniques to the preprocessedWeb log data, in order to discover useful patterns. As mentioned before, the most common and simple method that can be applied to such data is statistical analysis. More advanced data mining methods and algorithms tailored appropriately for use in the Web domain include association rules, sequential pattern discovery, clustering, and classification. Association rule mining is a technique for finding frequent patterns, associations, and correlations among sets of items. Association rules are used in order to reveal correlations between pages accessed together during a server session. Such rules indicate the possible relationship between pages that are often viewed together even if they are not directly connected, and can reveal associations between groups of users with specific interests. Aside from being exploited for business applications, such observations also can be used as a guide forWeb site restructuring, for example, by adding links that interconnect pages often viewed together, or as a way to improve the* *system's performance through prefetching Web data. Sequential pattern discovery is an extension of association rules mining in that it reveals patterns of cooccurrence incorporating the notion of time sequence. In theWeb domain such a pattern might be aWeb page or a set of pages accessed immediately after another set of pages. Using this approach, useful users' trends can be discovered, and predictions concerning visit patterns can be made. Clustering is used to group together items that have similar characteristics. In the context of Web mining, we can distinguish two cases, user clusters and page clusters. Page clustering identifies groups of pages that seem to be conceptually related according to the users' perception. User clustering results in groups of users that seem to behave similarly when navigating through a Web site. Such knowledge is used in e-commerce in order to perform market segmentation but is also helpful when the objective is to personalize a Web site. Classification is a process that maps a data item into one of several predetermined classes. In the Web domain classes usually represent different user profiles and classification is performed using selected features that describe each user's category. The most common classification algorithms are decision trees, na¨ıve Bayesian classifier, neural networks, and so on. After discovering patterns from usage data, a further analysis has to be conducted. The exact methodology that should be followed depends on the technique previously used. The most common ways of analyzing such patterns are either by using a query mechanism on a database where the results are stored, or by loading the results into a data cube and then performing OLAP operations. Additionally, visualization techniques are used for an easier interpretation of the results. Using these results in association with content and structure information concerning the Web site there can be extracted useful knowledge for modifying the site according to the correlation between user and content groups."*)(Pages 9-16, section 4); and returning the method bundle to the requestor *(i.e. "The purpose of Web usage mining is to reveal the knowledge hidden in the log files of a Web server. By applying statistical and data mining methods to the Web log data, interesting patterns concerning the users' navigational behavior can be identified, such as user and page clusters, as well as possible correlations between Web pages and user groups. TheWeb usage mining process can be regarded as a three-phase process, consisting of the data preparation, pattern discovery,* *and pattern analysis phases [Srivastava et al. 2000]. In the first phase, Web log data are preprocessed in order to identify users, sessions, pageviews and so on. In the second phase, statistical methods, as well as data mining methods (such as association rules, sequential pattern discovery, clustering, and classification) are applied in order to detect interesting patterns. These patterns are stored so that they can be further analyzed in the third phase of the Web usage mining process. A description of the fields included in a log entry of a Web usage log follows, along with a set of definitions of Web data abstractions, such as Web site, user, session, pageviews, and clickstreams. Technical issues concerning data preparation are discussed. A more detailed analysis of the methods employed in the Web usage mining process including simple log analysis is presented. Finally, a brief overview of the commercially available tools and applications specializing in log analysis or Web usage mining is given. Log analysis is regarded as the simplest method used in the Web usage mining process. The purpose of Web usage mining is to apply statistical and data mining techniques to the preprocessedWeb log data, in order to discover useful patterns. As mentioned before, the most common and simple method that can be applied to such data is statistical analysis. More advanced data mining methods and algorithms tailored appropriately for use in the Web domain include association rules, sequential pattern discovery, clustering, and classification. Association rule mining is a technique for finding frequent patterns, associations, and correlations among sets of items. Association rules are used in order to reveal correlations between pages accessed together during a server session. Such rules indicate the possible relationship between pages that are often viewed together even if they are not directly connected, and can reveal associations between groups of users with specific interests. Aside from being exploited for business applications, such observations also can be used as a guide forWeb site restructuring, for example, by adding links that interconnect pages often viewed together, or as a way to improve the system's performance through prefetching Web data. Sequential pattern discovery is an extension of association rules mining in that it reveals patterns of cooccurrence incorporating the notion of time sequence. In theWeb domain such a pattern might be aWeb page or a set of pages accessed immediately after another set of pages. Using this approach, useful users' trends can be discovered, and predictions concerning visit patterns can be made. Clustering is used to group together items that have* *similar characteristics. In the context of Web mining, we can distinguish two cases, user clusters and page clusters. Page clustering identifies groups of pages that seem to be conceptually related according to the users' perception. User clustering results in groups of users that seem to behave similarly when navigating through a Web site. Such knowledge is used in e-commerce in order to perform market segmentation but is also helpful when the objective is to personalize a Web site. Classification is a process that maps a data item into one of several predetermined classes. In the Web domain classes usually represent different user profiles and classification is performed using selected features that describe each user's category. The most common classification algorithms are decision trees, na¨ıve Bayesian classifier, neural networks, and so on. After discovering patterns from usage data, a further analysis has to be conducted. The exact methodology that should be followed depends on the technique previously used. The most common ways of analyzing such patterns are either by using a query mechanism on a database where the results are stored, or by loading the results into a data cube and then performing OLAP operations. Additionally, visualization techniques are used for an easier interpretation of the results. Using these results in association with content and structure information concerning the Web site there can be extracted useful knowledge for modifying the site according to the correlation between user and content groups."*)(Pages 9-16, section 4).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Erinaki to include a method comprising of receiving a request from a requestor to query a code portion for a method bundle including method metadata; searching the code portion for the method bundle; and returning the method bundle to the requestor with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claim 16, Boger teaches a method, wherein searching the code portion comprises searching in an instruction cache and further wherein, the code portion comprises an instruction code (i.e. *"In response to a request for data access descriptors by the optimizer instance, context 224 would determine whether data source caches existed for either of tables T1 and T2, which are accessed by query 202."*)(Column 15, lines 60-63).

7. Claims 3, 4, 6, 8, 9, 11, 15, 19, 20, 22, 25, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boger et al (U.S. Patent 6,996,556 and known hereinafter as Boger) in view of a non-patent literature titled "Web Mining for Web Personalization" by M. Eirinaki et al., ACM Transactions on Internet Technology, Vol. 3, No. 1, February 2003, pages 1-27 (and known hereinafter as Eirinaki) and in further in view of Adl-Tabatabai et al (U.S. Patent 6,317,869 and known hereinafter as Adl-Tabatabai).

As per claims 3, 19, and 25, Boger and Erinaki do not explicitly teach a method wherein the method metadata comprises a magic cookie having a bit pattern non-compliant with an instruction set architecture.

Adl-Tabatabai teaches a method wherein the method metadata comprises a magic cookie having a bit pattern non-compliant with an instruction set architecture (i.e. *"However, Java bytecodes have "ambiguous types" where the same variable may hold reference and non-reference values at different times during the execution of the method. A method is essentially a function or procedure in the program."* The preceding text clearly indicates that the non-compliant bit pattern is the non-reference value in the Java bytecodes.)(Column 6, lines 1-3).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Erinaki and with the further teachings of Adl-Tabatabai to include a method wherein the method metadata comprises a magic cookie having a bit pattern non-compliant with an instruction set architecture with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claims 4, 15, 20, 26, Boger and Erinaki do not explicitly teach a method, wherein storing the method metadata comprises storing the method metadata at an N-aligned address of the code portion.

Adl-Tabatabai teaches a method, wherein storing the method metadata comprises storing the method metadata at an N-aligned address of the code portion (i.e. *"The memory space 400 in FIG. 4a comprises of a run-time stack 410, registers 450, static variables (462, 468, 472) and objects (460, 464, 466, 470) on the heap. The run-time stack 410 may further comprise of a plurality of activation frames (420, 430, 440) for various program functions and methods. These activation frames are used as working space for functions and methods called during execution of the Java program. During program execution, numerous objects or variables may be declared and used within the program. An object, also referred to as a cell or node, is a run-time notion; any object is an instance of a certain class, created at execution time and made of a number of fields. An object may be a dynamically created class instance or an array or an individually allocated piece of data. An object is assumed to be a contiguous array of bytes or words, divided into fields. A field may contain a pointer or a non-pointer value."*)(Column 4, lines 58-67; column 5, lines 1-5).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Erinaki and with the further teachings of Adl-Tabatabai to include a method, wherein storing the method metadata comprises storing the method metadata at an N-aligned address of the code portion with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claims 6 and 11, Boger and Erinaki do not explicitly teach a method, wherein the code portion comprises compiled code for a method corresponding to the method metadata.

Adl-Tabatabai teaches a method, wherein the code portion comprises compiled code for a method corresponding to the method metadata (i.e. *"Computer programs are generally created as source code. The source code is then compiled into object code for execution. Programs generally exist as compiled object code in computer systems. The compiled code is usually designed to operate on only one particular operating system or on only one particular computer processor architecture. In order to use a certain program on several different computer architectures, the original source code must be compiled into object code for each different operating system and each different computer processor architecture."*)(Column 1, lines 18-27).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Erinaki and with the further teachings of Adl-Tabatabai to include a method, wherein the code portion comprises compiled code for a method corresponding to the method metadata with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claims 8 and 22, Boger teaches a method in querying the code portion (i.e. *"The primary task of a query optimizer is to choose the most efficient way to execute each database query, or request, passed to the database management system by a user." "An optimizer is often permitted to rewrite a query (or portion of it) into any equivalent form, and since for any given query there* *are typically many equivalent forms, an optimizer has a countably infinite universe of extremely diverse possible solutions (plans) to consider."*)(Column 1, lines 63-66; Column 2, lines 17-22).

Boger and Erinaki do not explicitly teach a method, wherein the code portion comprises searching at N-aligned addresses of the code portion.

Adl-Tabatabai teaches a method, wherein the code portion comprises searching at N-aligned addresses of the code portion (i.e. *"The memory space 400 in FIG. 4a comprises of a run-time stack 410, registers 450, static variables (462, 468, 472) and objects (460, 464, 466, 470) on the heap. The run-time stack 410 may further comprise of a plurality of activation frames (420, 430, 440) for various program functions and methods. These activation frames are used as working space for functions and methods called during execution of the Java program. During program execution, numerous objects or variables may be declared and used within the program. An object, also referred to as a cell or node, is a run-time notion; any object is an instance of a certain class, created at execution time and made of a number of fields. An object may be a dynamically created class instance or an array or an individually allocated piece of data. An object is assumed to be a contiguous array of bytes or words, divided into fields. A field may contain a pointer or a non-pointer value."*)(Column 4, lines 58-67; column 5, lines 1-5).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Erinaki and with the further teachings of Adl-Tabatabai to include a method, wherein the code portion comprises searching at N-aligned addresses of the code portion with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claim 9, Boger teaches a method wherein storing the method metadata comprises storing the method metadata (i.e. *"The embodiments discussed hereinafter utilize a metadata manager to facilitate the collection of metadata from a database, and to supply such metadata* *to a plurality of optimizer instances in connection with the optimization of database queries." "As an additional matter, by implementing metadata handling functionality outside of optimizer instances, improvements to metadata collection, refinement and maintenance algorithms may be implemented within a metadata manager without requiring modification of optimizer program code."*)(Column 4, lines 50-54; lines 64-67; column 5, lines 1-2).

Boger and Erinaki do not explicitly teach a method, wherein storing the method metadata at an opposite side of a boundary location at an N-aligned address of the code portion at which a basic block is stored.

Adl-Tabatabai teaches a method, wherein storing the method metadata at an opposite side of a boundary location at an N-aligned address of the code portion at which a basic block is stored (i.e. *"The memory space 400 in FIG. 4a comprises of a run-time stack 410, registers 450, static variables (462, 468, 472) and objects (460, 464, 466, 470) on the heap. The run-time stack 410 may further comprise of a plurality of activation frames (420, 430, 440) for various program functions and methods. These activation frames are used as working space for functions and methods called during execution of the Java program. During program execution, numerous objects or variables may be declared and used within the program. An object, also referred to as a cell or node, is a run-time notion; any object is an instance of a certain class, created at execution time and made of a number of fields. An object may be a dynamically created class instance or an array or an individually allocated piece of data. An object is assumed to be a contiguous array of bytes or words, divided into fields. A field may contain a pointer or a non-pointer value."*)(Column 4, lines 58-67; column 5, lines 1-5).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Erinaki and with the further teachings of Adl-Tabatabai to include a method, wherein storing the method metadata at an opposite side of a boundary location at an N-aligned address of the code portion at which a basic block is stored with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

8. Claims 17 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boger et al (U.S. Patent 6,996,556 and known hereinafter as Boger) in view of a non-patent literature titled "Web Mining for Web Personalization" by M. Eirinaki et al., ACM Transactions on Internet Technology, Vol. 3, No. 1, February 2003, pages 1-27 (and known hereinafter as Eirinaki) and in further view of Buhrke et al (U.S. Patent 5,806,029 and known hereinafter as Buhrke).

As per claim 17, Boger and Eirinaki do not explicitly teach a method, wherein searching the code portion comprises bidirectionally searching the code portion for the method bundle.

Buhrke teaches a method, wherein searching the code portion comprises bidirectionally searching the code portion for the method bundle (i.e. *"This is even more problematic for an N-best decoding scheme using a bidirectional search where the backward search should maintain the context dependency used in the forward search in order to achieve search consistency."*)(Column 4, lines 2-6).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Erinaki and with the further of Buhrke to include a method, wherein searching the code portion comprises bidirectionally searching the code portion for the method bundle with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

As per claim 28, Boger and Erinaki do not explicitly teach a system wherein the memory further comprises instructions that if executed enable the system to search for the method metadata using one of a forward search, a backward search, or a bidirectional search.

Buhrke teaches a system wherein the memory further comprises instructions that if executed enable the system to search for the method metadata using one of a forward search, a backward search, or a bidirectional search (i.e. *"This is even more problematic for an N-best decoding scheme using a bidirectional search where the backward search should maintain the context dependency used in the forward search in order to achieve search consistency."*)(Column 4, lines 2-6).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Boger with the teachings of Erinaki and with the further teachings of Buhrke to include a system wherein the memory further comprises instructions that if executed enable the system to search for the method metadata using one of a forward search, a backward search, or a bidirectional search with the motivation to facilitate the management of metadata (Boger, column 3, line 52).

Conclusion

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Farhan M. Syed whose telephone number is 571-272-7191. The examiner can normally be reached on 8:30AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin can be reached on 571-272-4146. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JEFFREY GAFFIN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

FMS